UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/715,772	11/17/2000	Jack B. Dennis	84781-US2	7033

26111 7590 12/08/2006

STERNE, KESSLER, GOLDSTEIN & FOX PLLC
1100 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
DANG, KHANH	
ART UNIT	PAPER NUMBER
2111	

DATE MAILED: 12/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/715,772
Filing Date: November 17, 2000
Appellant(s): DENNIS ET AL.

Messinger
For Appellant

SUPPLEMENTAL EXAMINER'S ANSWER

This is in response to the reply brief filed 10/02/2006.

NOTICE TO APPLICANTS

This application, assigned to and examined by Ex. Justin, is now assigned to Ex. Dang. Any future contact should be directed to Ex. Dang whose contact information is provided at the end of this Office Action.

Further, Applicants are reminded that according to MPEP 1214.04, in situation when the application has been transferred or assigned to an examiner other than the one who rejected the claims leading to the appeal , the second examiner should give full faith and credit to the prior examiner's search. However, If the examiner has specific knowledge of the existence of a particular reference or references which indicate non-patentability of any of the appealed claims as to which the examiner was reversed, the Examiner should submit the matter to the Technology Center (TC) Director for authorization to reopen prosecution under **37 CFR 1.198** for the purpose of entering the new rejection. See **MPEP § 1002.02(c)** and **MPEP § 1214.07**. The TC Director's approval is placed on the action reopening prosecution.

Supplemental Examiner's Answer:

The Examiner's Answer, **section (8)** does not list the patents relied upon in the rejection as evidenced.

A list of the patents relied upon in the rejections is provided below:

The following are the related appeals, interferences, and judicial proceedings known to the examiner which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

(8) Evidence Relied Upon

5,815,727	Motomura	9/98
5,418,917	Hiraoka et al.	5/95
5,938,765	Dove et al.	8/99

Response to Reply Brief:

It is first noted that many if not all of Applicants' arguments presented in this Reply Brief as well as the Appeal Brief have already been addressed in the Examiner's Answer prepared by Ex. King.

The followings are only observations directly resulted from a review of Applicants' Reply Brief and the Examiner's Answer during a process of preparing a response to Applicants' Reply Brief. No new patent search has been made.

Applicants argue that "unlike Motomura, in an embodiment of the claimed invention, the earlier thread does not generally enter a waiting state. The execution of a 'wait' or 'no wait' instruction by a thread in a processing slice is relevant only to a peripheral operation. (Specification, p. 9, ll. 13-15; p. 11, ll. 6-16). During the execution of a peripheral operation (even a 'wait' operation), instructions may be dispatched and processed concurrently, on a common clock cycle." See Reply Brief, page 2.

At the outset, it is acknowledged that Applicants concede that "waiting state" is used in the claimed invention.

As clearly stated in the originally filed specification, page 9, lines 21-28, "if the command message is a wait message instruction, the issuing thread is **stalled** **for an interval** during which the responding peripheral unit carries out the peripheral operation. **During this interval, the resources associated with the issuing thread are available to other threads in the issuing slice**. In this way, high resource utilization can be achieved. If it is a no_wait instruction, the issuing thread continues executing its sequence without waiting for the peripheral operation to be completed. The issuing thread may or may not need a response from the peripheral unit."

It is also important to note that independent claims 1, 14, 27, 40, and 41 require that "the processing slice executes the instructions from more than one of the plurality of threads concurrently, in a clock cycle." In another word, instructions from more than one threads are executed concurrently in a clock cycle.

From the above, it is clear that the term "executed concurrently" does NOT mean that there is no "wait state" or "stalled" state between threads execution in Applicants' claimed invention. Rather, as discussed above, with the use of "waiting state" and "stalled states" between threads execution, the threads can be said to be "executed concurrently, in a clock cycle." The term "in a clock cycle" is a key term here, and it is clear that the word "concurrently" means that the threads are executed all within a time frame or specifically, "a clock cycle."

Applicants argue that "Motomura, on the other hand, does not execute the instructions from more than one thread concurrently in a clock cycle. On the contrary, each processor in Motomura can execute only one thread at a time, and it is disclosed that each thread must go into a 'waiting' or 'completed' state before another thread is assigned to the processor. (Motomura, col. 8, ll. 40-51)."

As discussed above, while the Examiner agrees with Applicants that "waiting" or "completed" states may be introduced between execution of threads, but the same thing ("wait state" or "stalled" thread) is also employed in multithreading disclosed by Applicants. In Motomura, parallel processing and multithreading are employed. By definition, "Parallel computing is the <u>simultaneous execution of the same task</u> (split up and specially adapted) on multiple processors in order to obtain results faster. The idea is based on the fact that the process of solving a problem usually can be divided into smaller tasks, which may be carried out simultaneously with some coordination. See definition of Parallel Computing from Wikipedia,

<ttp://en.wikipedia.org/wiki/Parallel_computing>

As a mater of fact, Motomura, in the abstract, discloses "a parallel processor system executing a program consisted of a plurality of threads in parallel per threads."

Further, multithreading, by definition, is "Multiple threads can be executed in parallel on many computer systems. On a multiprocessor system, threading can be achieved via multiprocessing, wherein different threads and processes can run literally <u>simultaneously</u> on different processors. This *multithreading* generally occurs by time slicing, wherein a single processor switches between different threads, in which case the processing is not literally simultaneous, for the single processor is only really doing one thing at a time. This switching can happen so fast as to give the illusion of simultaneity to an end user. For instance, a typical PC today contains only one processor core, but you can run multiple programs at once, such as a word processor alongside an audio playback program; though the user experiences these things as simultaneous, in truth, the processor is quickly switching back and forth between these separate processes. On a multiprocessor system, threading can be achieved via multiprocessing, wherein different threads and processes can run literally simultaneously on different processors. See definition of Thread from Wikipedia,

<http://en.wikipedia.org/wiki/Multithreaded>

Thus, it is clear that in Motomura, the instructions from more than one of the plurality of threads are executed concurrently, in a clock cycle.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Khanh Dang

Khanh Dang
Primary Examiner